Normalization

Functional Dependency: Relationship between the attributes of a relation (table).
Determinant: Left hand-side of the functional dependency.

Example: Let's find Super Key(s), Candidate Key(s), Primary Key of table R.

R(a, b, c, d, e, f)

Functional Dependency (FD):

1. a,b→c,e
2. c→f
3. f→d
4. d→a,b

Solution:

Step 1: Compute the closures of all FDs and determine which are SKs.

1. $\{a,b\}^+ = \{a,b,c,e,f,d\}$ ✓S.K.
2. $\{c\}^+ = \{c,f,d,a,b,e\}$ ✓S.K.
3. $\{f\}^+ = \{f,d,a,b,c,e\}$ ✓S.K.
4. $\{d\}^+ = \{d,a,b,c,e,f\}$ ✓S.K.

Step 2: Find the Candidate Keys (minimal SK)

~~a,b (2)~~ X CK
c (1) ✓ CK
f (1) ✓ CK
d (1) ✓ CK

Step 3: Choose the PK.

{c}, {f} or {d} →only one of them has to be the PK. Remember that every relation must have a PK.

Example:

StudentCourse

Stdid	stdName	Ccode	Ctitle	Credits	letterGrade
101	Ali	ITEC212	DBMS	4	A
102	Ayşe	ITEC243	OOP	4	B
103	Fatma	ITEC212	DBMS	4	C
101	Ali	ITEC309	NW	3	C
102	Ayşe	ITEC212	DBMS	4	D
..	..	..	..	..	..

List down all reasonable functional dependencies.

1. Stdid→stdName
2. Ccode→ctitle, credits
3. Stdid,ccode→letterGrade

Let's normalize table StudentCourse

Step 1:

Find SK(s) of StudentCourse table.

1. {stdid}$^+$={stdid, stdName} **X SK**
2. {ccode}$^+$={ccode, ctitle, credits} **X SK**
3. {stdid,ccode}$^+$={ stdid, ccode, letterGrade, stdName, ctitle, credits} ✓ **SK**

Table StudentCourse is not in BCNF since FD 1 and FD2 are not Super Keys.

Step 2:

Rule: Choose any violating F.D. and form a new table from its closure. List all F.D.s that valid on the new table. Find the Primary Key.

Let's pick FD2:

R1(ccode, ctitle, credits)

FD:

2. ccode→ctitle, credits ✓SK. Hence PK.

No violations!!! **Table R1 is in BCNF**

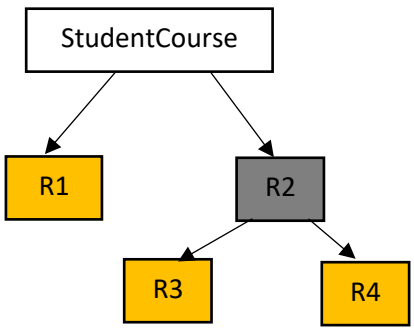
Step 3:

Rule: Eliminate all the non key attributes of the table formed in Step2 from the table that you normalize. Form a new table from the remainders of the main table. List all F.D.s that are valid on the new table.

R1(ccode, ctitle, credits)

StudentCourse(stdid, stName, ccode, ~~ctitle~~, ~~credits~~, lettergrade)

Eliminate non-key attributes (ctitle and credits) of R1 from StudentCourse (main) table. Let's form a new table form the remainders of StudentCourse table.



R2(stdid, stName, ccode, lettergrade)

FD:

1. stdid→stdName **XSK (violation for BCNF)**
3. stdid,ccode→lettergrade ✓**SK**

Table R2 is not in BCNF. Normalize it.

Now, we need to repeat step 2 and step 3 for table R2 until it is in BCNF!

Normalize R2

Step2:

We have only one FD that violate BCNF (which is not a SK) for R2. So we need to pick that FD and start normalizing the table.

We pick FD1 (of R2)

R3(<u>stdid</u>, stdName)
FD:

1. stdid→stdName ✓**SK. Hence PK.**

Table R3 is in BCNF.

Step3:

Our main table is → R2(stdid, ~~stdName~~, ccode, lettergrade)
The table we formed in Step2 → R3(<u>stdid</u>, stdName)

Let's form a new table from the remainders of the main table.

R4(<u>stdid</u>, <u>ccode</u>, lettergrade)
FD:

3. stdid,ccode→lettergrade ✓**SK. Hence PK.**

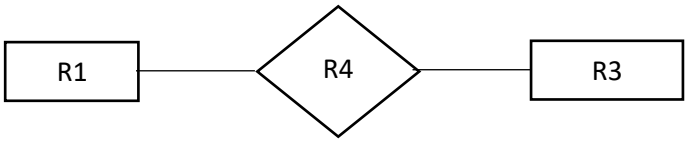
Table R4 is in BCNF.

<u>List of Normalized Tables:</u>

R1(<u>ccode</u>, ctitle, credits)

R3(<u>stdid</u>, stdName)

R4(<u>stdid</u>, <u>ccode</u>, lettergrade)



Example 2:

R(projId, projName, empId, empName, jobcode, jobTitle, dateAssigned)

FDs.

1. projId→projName
2. empId→empName, jobCode
3. jobCode→jobTitle
4. projId, empId→dateAssigned

Normalize table up to BCNF.

Step1:

1. $\{projId\}^+$=$\{projId, projName\}$ **X SK**
2. $\{empId\}^+$= $\{empId, empName, jobCode, jobTitle\}$ **X SK**
3. $\{jobCode\}^+$=$\{jobCode, jobTitle\}$ **X SK**
4. $\{projId, empId\}^+$=$\{projId, empId, dateAssigned, projName, empName, jobCode, jobTitle\}$ ✓**SK**

Table R is not in BCNF!!!! We need to normalize it.

Let's normalize Table R

Step2:

Take FD 2.

R1(empId, empName, jobCode, jobTitle)

FD.

2. empId→empName, jobCode ✓**SK. PK**
3. jobCode→jobTitle **X SK**

R1 is not in BCNF!!!

Step3:

R(projId, projName, empId, ~~empName~~, ~~jobcode~~, ~~jobTitle~~, dateAssigned)

R1(empId, empName, jobCode, jobTitle)

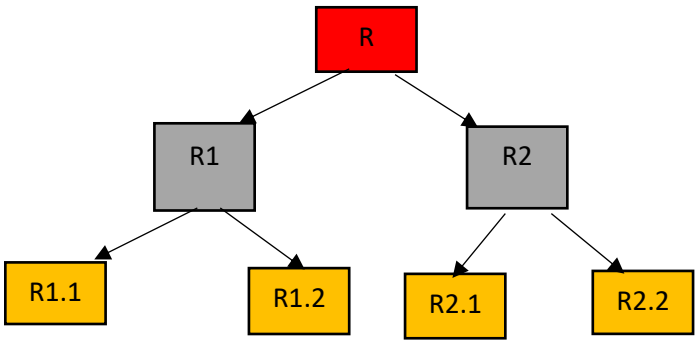
New table:

R2(projId, projName, empId, dateAssigned)

FD.

1. projId→projName **X SK**
4. projId, empId→dateAssigned ✓**SK.**

R2 is not in BCNF!!!



Normalizing R1

Step 2:

R1.1(jobCode, jobTitle)

FD.

3. jobCode→jobTitle ✓**SK. PK**

R1.1 is in BCNF.

Step 3:

MAIN table R1(empId, empName, jobCode, ~~jobTitle~~)

R1.2(empId, empName, jobCode)

FD.

2. empId→empName, jobCode ✓**SK**

R1.2 is in BCNF.

Normalizing R2

Step 2:

R2.1(projId, projName)

FD.

1. projId→projName ✓**SK**

Table R2.1 is in BCNF.

Step 3:

MAIN table R2(projId, ~~projName~~, empId, dateAssigned)

R2.2(projId, empId, dateAssigned)

4. projId, empId→dateAssigned ✓**SK**

Table R2.2 is in BCNF.

List of Normalized Tables:

R1.1(jobCode, jobTitle)

R1.2(empId, empName, jobCode)

R2.1(projId, projName)

R2.2(projId, empId, dateAssigned)